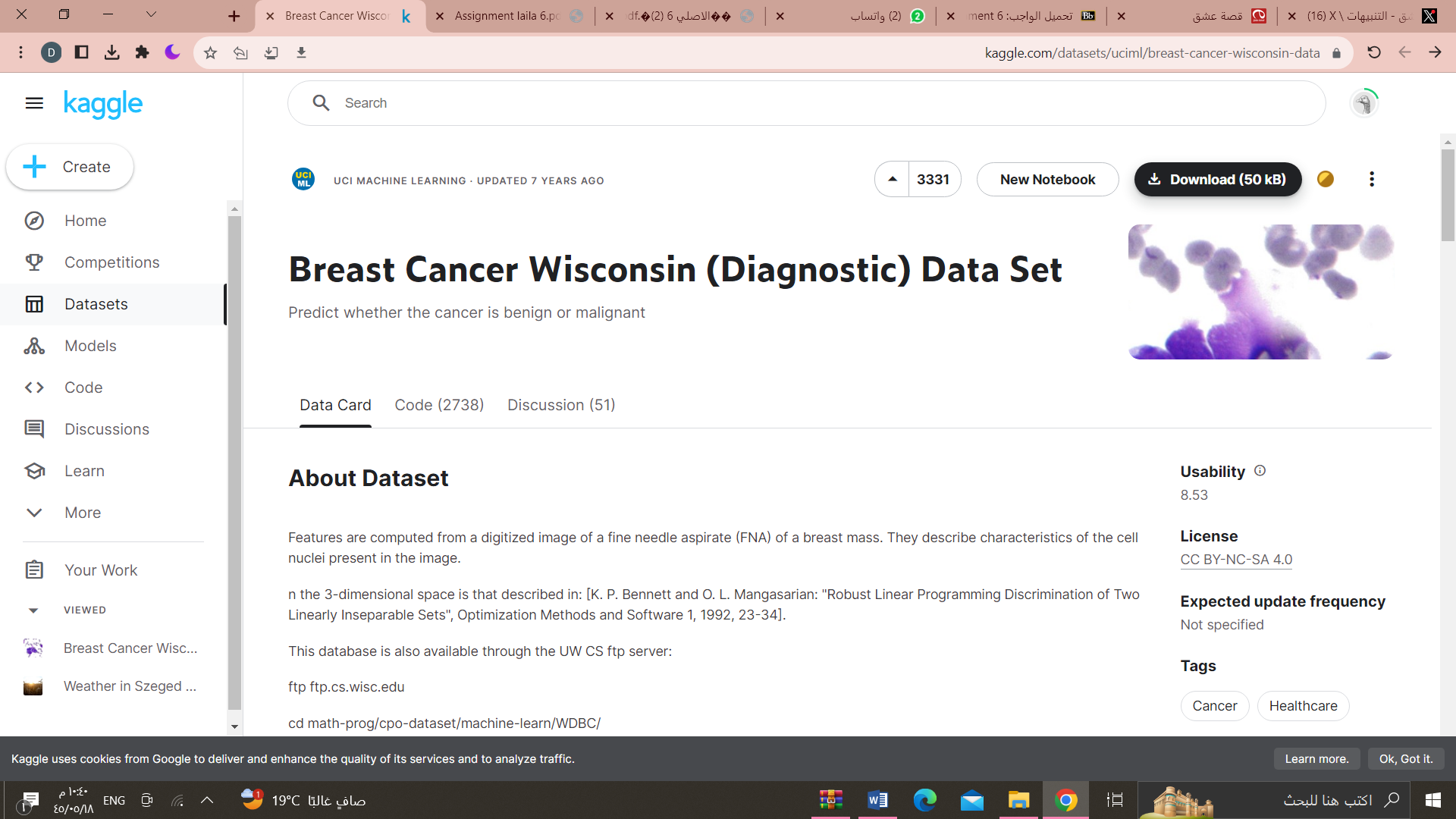
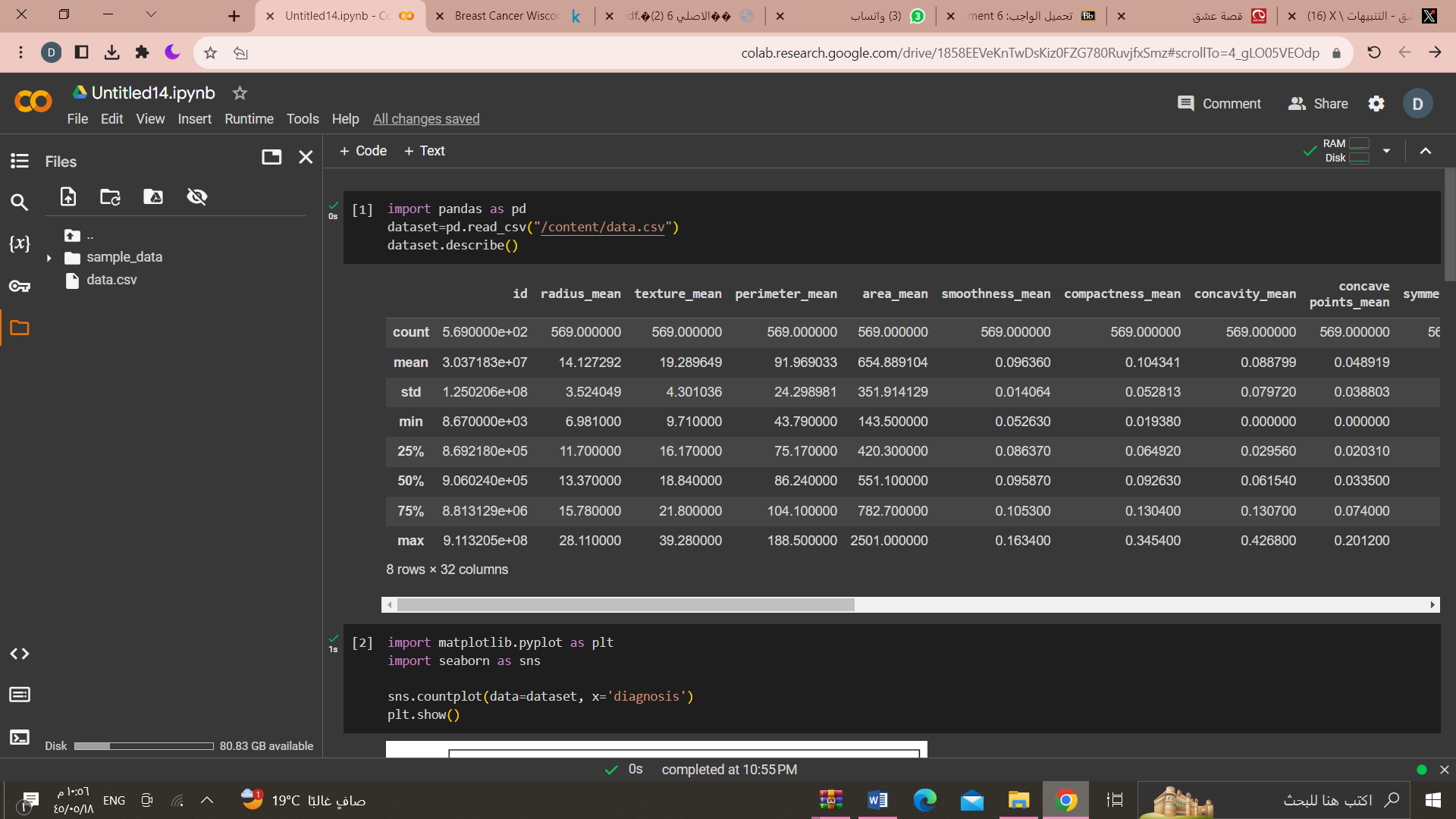
|  |  |  |
| --- | --- | --- |
| Assignment 6 | | |
| Classification | | |
| Student Name: saja eid aldosary | Section No: | Student ID :442050483 |

Use the breast cancer dataset at the link <https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data> to build a classification model using the data provided.

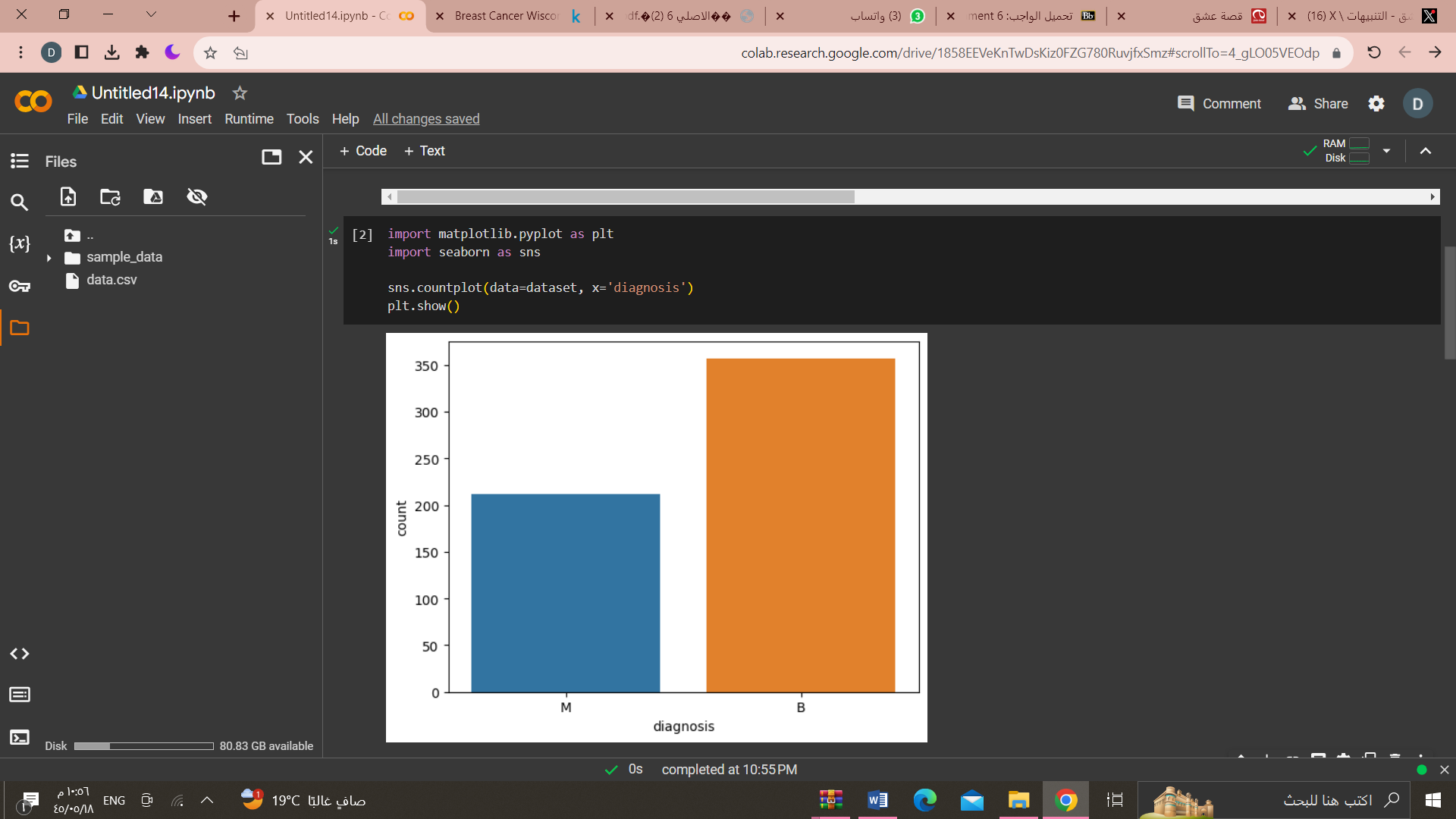
* Download dataset



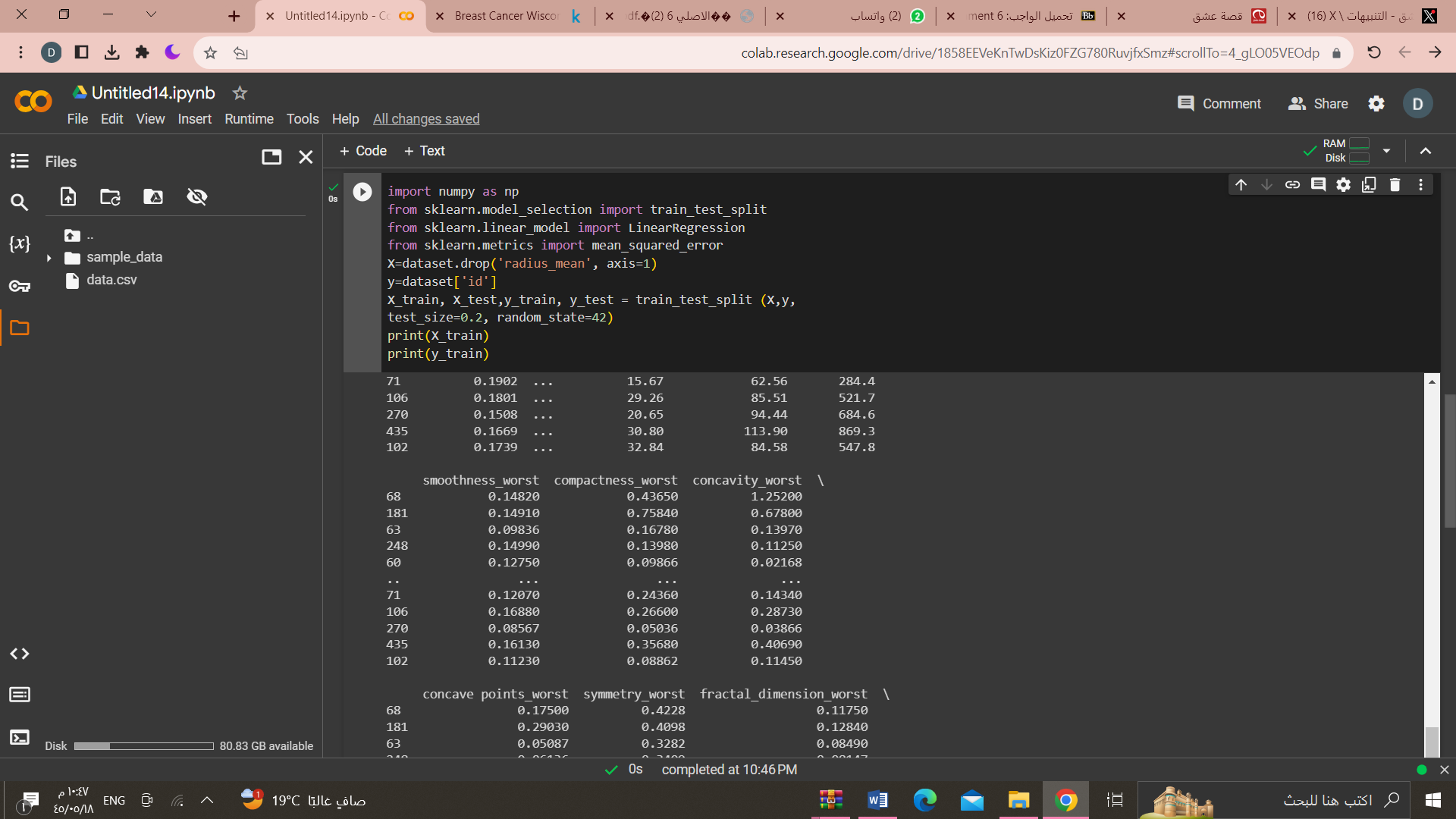
* Read the dataset
* import pandas as pd
* dataset=pd.read\_csv("/content/data.csv")
* dataset.describe()

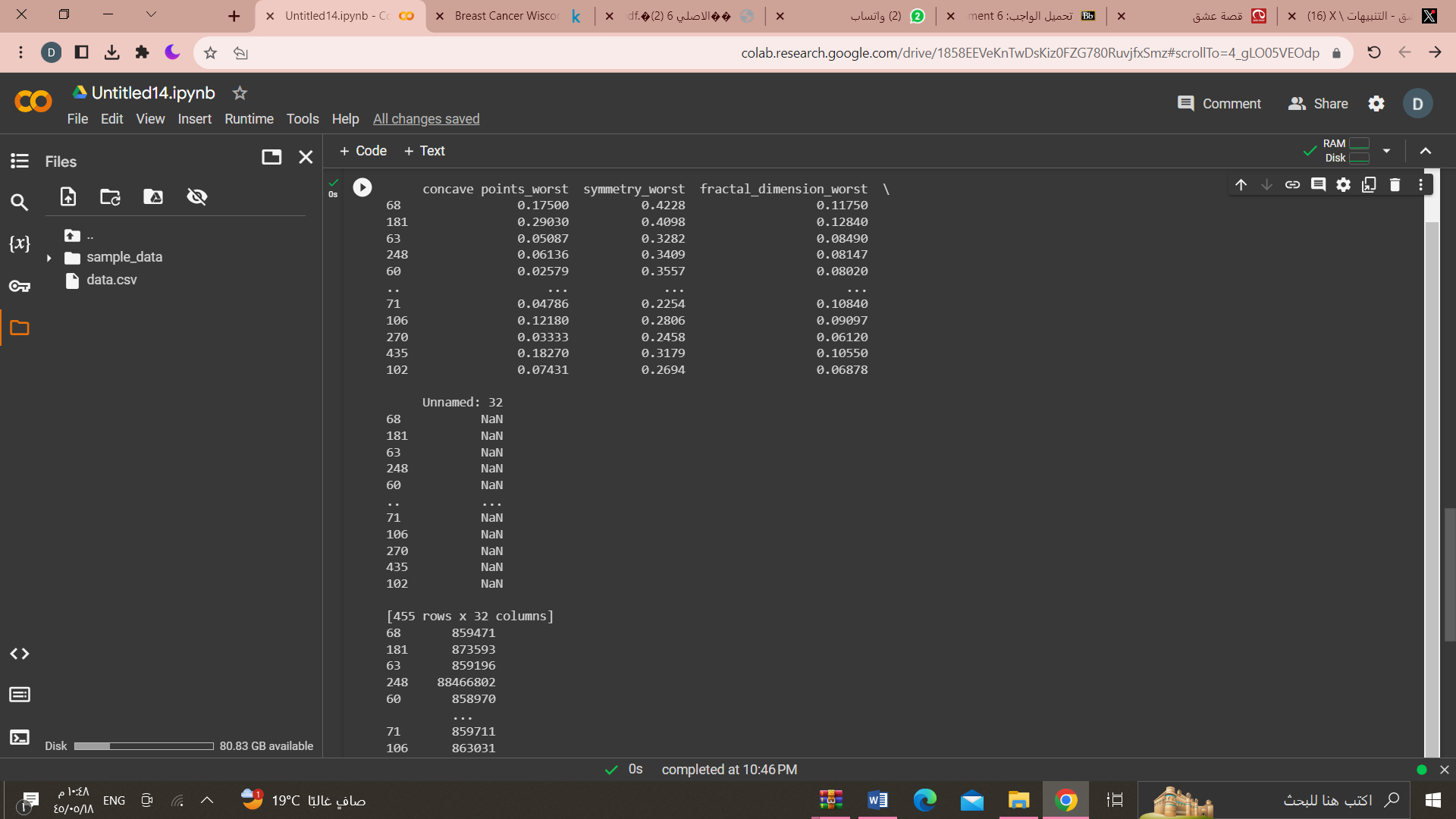


* Use visualization if possible.
* import matplotlib.pyplot as plt
* import seaborn as sns
* sns.countplot(data=dataset, x='diagnosis')
* plt.show()

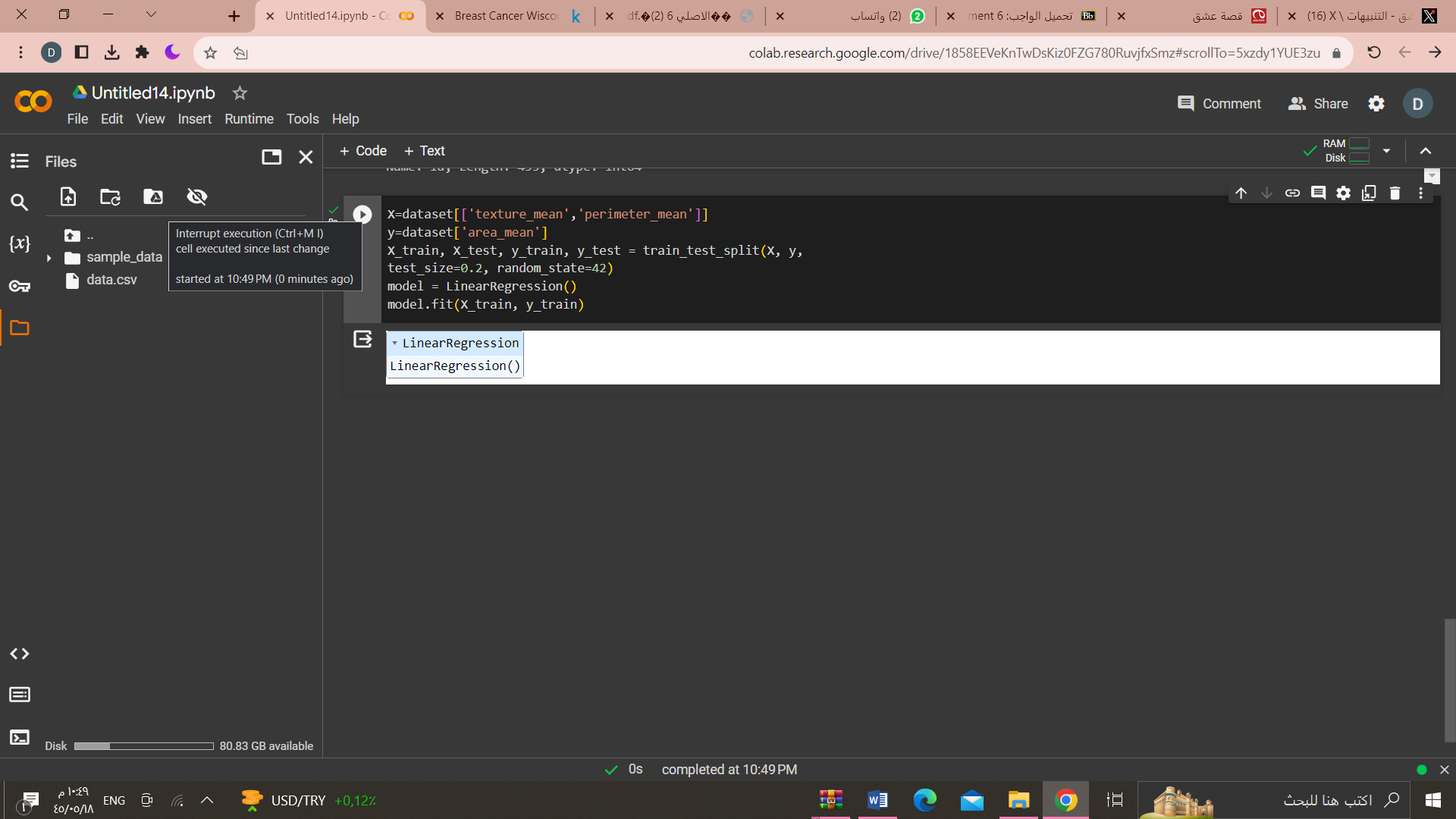


* Split the dataset 80% for train-20% for test
* import numpy as np
* from sklearn.model\_selection import train\_test\_split
* from sklearn.linear\_model import LinearRegression
* from sklearn.metrics import mean\_squared\_error
* X=dataset.drop('radius\_mean', axis=1)
* y=dataset['id']
* X\_train, X\_test,y\_train, y\_test = train\_test\_split (X,y,
* test\_size=0.2, random\_state=42)
* print(X\_train)
* print(y\_train)

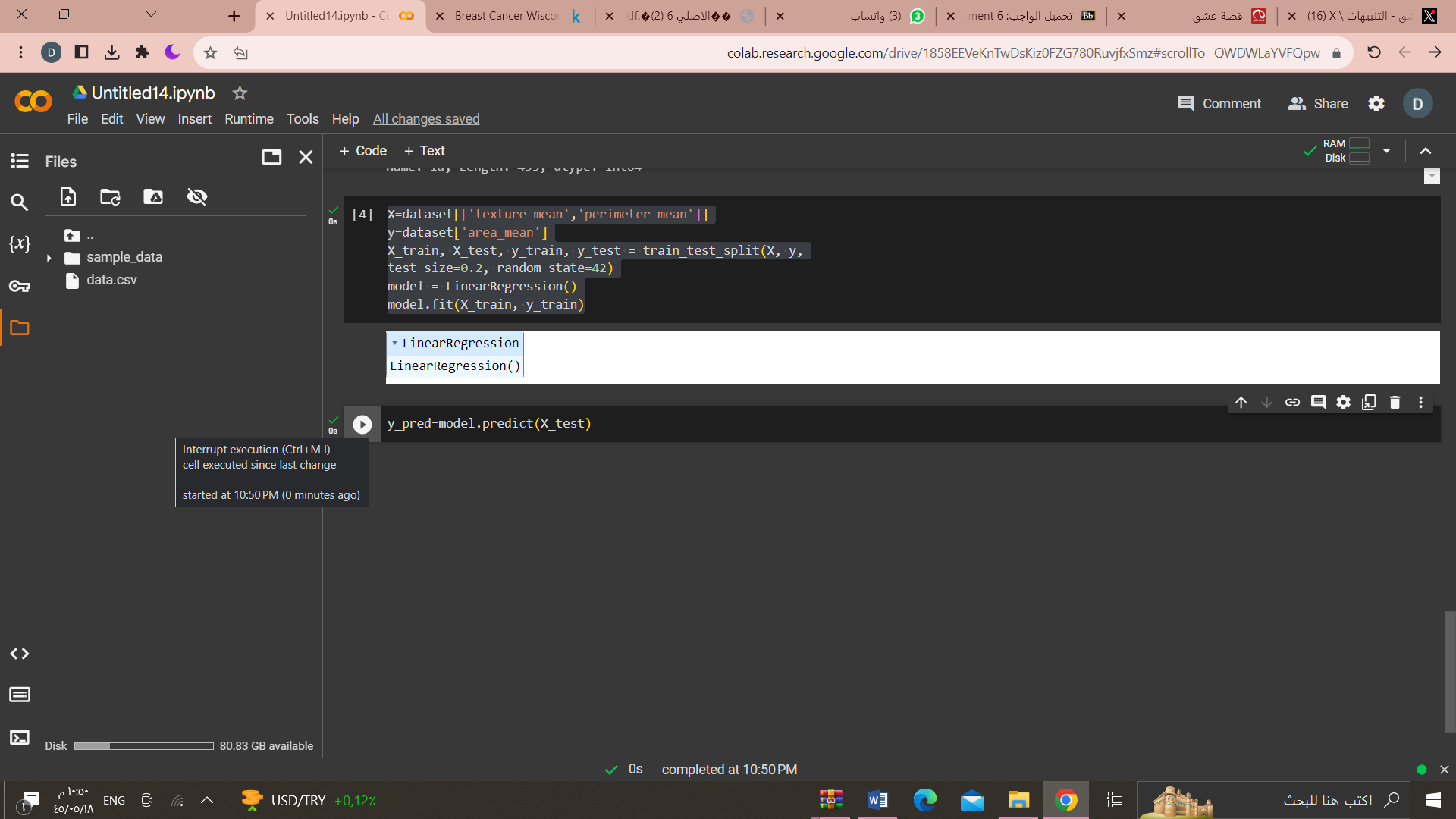




* Build the model using the train dataset split.
* X=dataset[['texture\_mean','perimeter\_mean']]
* y=dataset['area\_mean']
* X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,
* test\_size=0.2, random\_state=42)
* model = LinearRegression()
* model.fit(X\_train, y\_train)

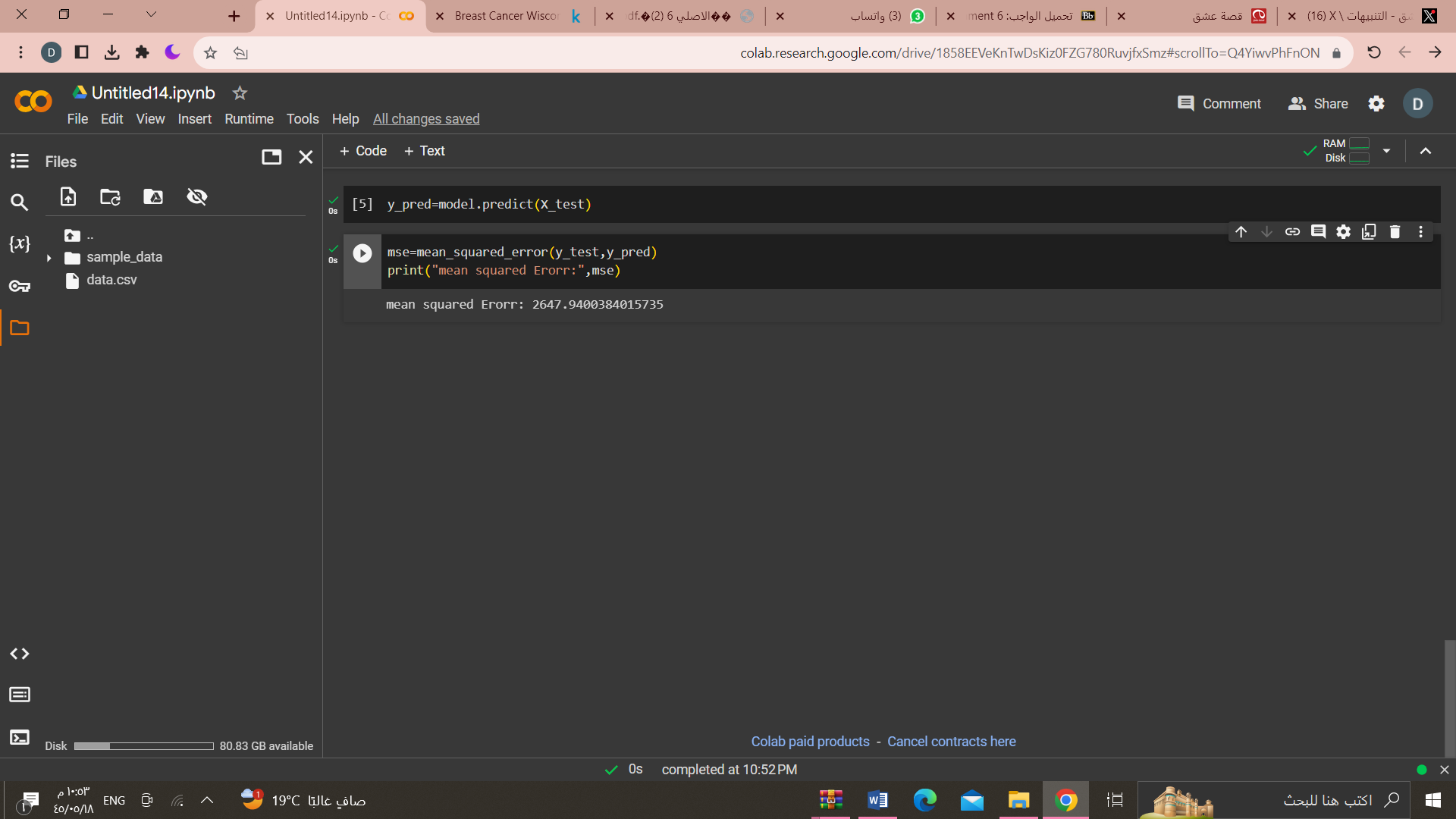


* Use the test dataset for evaluation.
* y\_pred=model.predict(X\_test)



mse=mean\_squared\_error(y\_test,y\_pred)

print("mean squared Erorr:",mse)



* Show the performance of the model.
* model.score(X\_train,y\_train)
* model.score(X\_test, y\_test)

